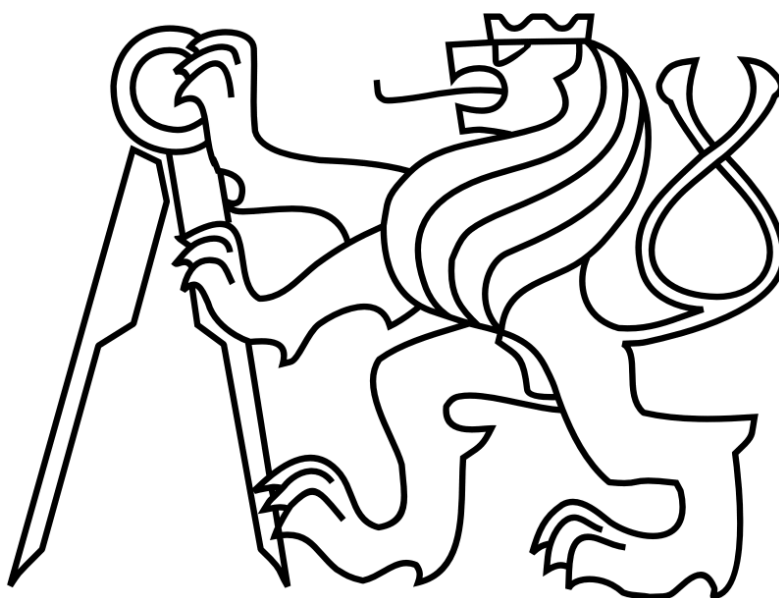


CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

# Bachelor thesis



Michal Zeman

**Application for Feature Extraction  
from Eye Movement Signal Analysis**

**2013**

**Department of Cybernetics**

Thesis supervisor: **Ing. Martin Macaš, Ph.D.**

## BACHELOR PROJECT ASSIGNMENT

**Student:** Michal Z e m a n

**Study programme:** Cybernetics and Robotics

**Specialisation:** Robotics

**Title of Bachelor Project:** Application for Feature Extraction from Eye Movement Signal Analysis

### Guidelines:

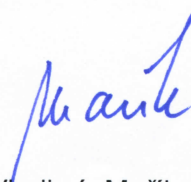
1. Develop an application with GUI for automatic feature extraction from results of analysis of eye movement signals.  
The application will enable:
  - a. mass feature extraction for multiple analysis files
  - b. basic statistical analysis and visualization of features
  - c. simple extendibility in terms of additional feature extractors
2. Validate the application on provided analysis files
  - a. Apply the mass feature extraction
  - b. Compare the results with the sample results provided by the supervisor

### Bibliography/Sources:

- [1] Jakub Snopek: Metody analýzy záznamů očních pohybů při čtení a v sekvenčních úlohách. Diplomová práce, katedra kybernetiky, ČVUT, 2003.
- [2] Martin Macaš: Dyslexia detection using artificial neural networks. Katedra kybernetiky, ČVUT, 2005.

**Bachelor Project Supervisor:** Ing. Martin Macaš, Ph.D.

**Valid until:** the end of the winter semester of academic year 2013/2014

  
prof. Ing. Vladimír Mařík, DrSc.  
**Head of Department**



  
prof. Ing. Pavel Ripka, CSc.  
**Dean**

Prague, January 10, 2013

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Student:** Michal Z e m a n

**Studijní program:** Kybernetika a robotika (bakalářský)

**Obor:** Robotika

**Název tématu:** Aplikace pro extrakci příznaků z analýz signálů očních pohybů

### Pokyny pro vypracování:


1. Vytvořte aplikaci s GUI pro automatickou extrakci příznaků z výsledků analýz signálů očních pohybů.  
Aplikace bude umožňovat:
  - a. hromadnou extrakci příznaků pro více souborů analýz
  - b. základní statistickou analýzu a vizualizaci příznaků
  - c. snadnou rozšiřitelnost o další extraktory
2. Validujte aplikaci na souborech analýz dodaných vedoucím práce
  - a. Provedte hromadnou extrakci příznaků
  - b. Porovnejte s dodaným vzorovým výsledkem

### Seznam odborné literatury:

- [1] Jakub Snopek: Metody analýzy záznamů očních pohybů při čtení a v sekvenčních úlohách. Diplomová práce, katedra kybernetiky, ČVUT, 2003.
- [2] Martin Macaš: Dyslexia detection using artificial neural networks. Katedra kybernetiky, ČVUT, 2005.

**Vedoucí bakalářské práce:** Ing. Martin Macaš, Ph.D.

**Platnost zadání:** do konce zimního semestru 2013/2014

  
prof. Ing. Vladimír Mařík, DrSc.  
**vedoucí katedry**



  
prof. Ing. Pavel Ripka, CSc.  
**děkan**


V Praze dne 10. 1. 2013



## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne.....24.5.2013.....

  
.....

## Acknowledgements

First and foremost, I would like to thank my supervisor Ing. Martin Macaš for many valuable suggestions, moral support and productive conversations. My gratitude also goes to Ing. Daniel Novák, who introduced me to Ing. Macaš. Finally, let me thank my whole family for their great support and great patience.

### *Abstrakt*

Hlavním cílem této práce bylo navrhnout aplikaci s grafickým uživatelským rozhraním pro hromadnou extrakci příznaků z analýz signálů očních pohybů v prostředí Matlab a struktury souborů příznakových extraktorů. Tato aplikace významně pomůže v budoucím studiu očních pohybů. Pomocí videookulografické metody (VOG) byla naměřena data pro otestování hromadné extrakce příznaků. Bylo vyhodnoceno celkem 15 záznamů pocházejících z měření očních pohybů zdravých subjektů při řešení verbálních i neverbálních sekvenčních úloh. Na tomto vzorku dat byla otestována funkčnost automatické hromadné extrakce příznaků.

### *Abstract*

The main goal of this work is to propose the application with graphic user interface for mass feature extraction from eye movement signal analysis in Matlab environment and structure of feature extractors files. This application will significantly help in future studies of eye movements. Using the videooculography method (VOG) were measured the signal data for mass feature extraction. In total signal data of 15 healthy subjects in dealing with verbal and non-verbal sequential tasks were processed. On this data sample was tested the functionality of automatic mass feature extraction.

# Contents

1	Introduction	1
1.1.	Main goals	2
1.2.	Motivation	2
1.3.	Recording of eye movements	3
1.4.	Signal analysis	4
1.4.1.	Classification of eye movements	5
1.5.	Feature extraction	6
1.5.1.	Position based features	6
1.5.2.	Numerosity based features	6
1.5.3.	Latency and distance based features	7
1.5.4.	Frequency based features	7
1.6.	Classification	8
2	Implementation	9
2.1.	Signal data	9
2.1.1.	EMSA structure	9
2.2.	Feature extractors	12
2.3.	Graphical User Interface	12
2.3.1.	Output	14
3	Validation	15
3.1.	Feature extractors description	15
3.1.1.	Out of screen proportion (out_of_screen, out_of_screen2)	15
3.1.2.	Stimuli type (stimuli_type)	17
3.1.3.	Task time (task_time)	17
3.2.	Mass extraction	18
4	Conclusion	20
4.1.	Future work	21
A	Eye Movements Feature Extraction Tool manual	22
A.1	Table of contents	23

## List of tables

Table 1 - This table shows different types of eye movements, their length in milliseconds, amplitude and velocity (adapted from Holmqvist, 2011).	5
Table 2 - extractors	18
Table 3 - file_names	19
Table 4 - results	19

## List of figures

Figure 1 - A block diagram of the proposed method, the part highlighted in yellow is concerned with this bachelor thesis	2
Figure 2 - EMSA structure	10
Figure 3 - 33.4% of signal data are out of screen	16
Figure 4 - dots2	17
Figure 5 - Task time of different subjects	18

## List of pictures

Picture 1 - I4Tracking measuring device	4
Picture 2 - EyeMove Toolbox	4
Picture 3 - Saccades and fixations during reading task (picture adapted from Rayner, 2007)	6
Picture 4 - Application window	13

## Software

Matlab Version 2012b, The MathWorks, Inc.  
I4Tracking, Medicton s.r.o.  
EMSA toolbox



# Chapter 1

## Introduction

Human eye is an extraordinary organ. The ability to see helped the development of human and other species. In last few decades it happened to be not only our main tool to see, but also a powerful tool in diagnose of serious human conditions. The measurement of human eye movements in last few decades led to important discoveries about psychological processes that occur during reading, visual search and scene perception.

There are several studies describing the relationship between eye movements and genetic or developmental disorders such as dyslexia (Pavlidis, 1985; Biscaldi, 1998; Macaš, 2005), sexual deviance, schizophrenia (O'Driscoll and Callahan, 2008) etc. However these studies do not provide any clear unified conclusions. In a study of eye movements is a lot of space for additional research.

This work focus on providing user application that can help in further research of eye movements using the methods of artificial intelligence and statistical analysis. The system of eye movement data acquisition and their processing used in this work consist of four separated main blocks (Figure 1). The connections between them tell us that output of one block serves as input into another block.

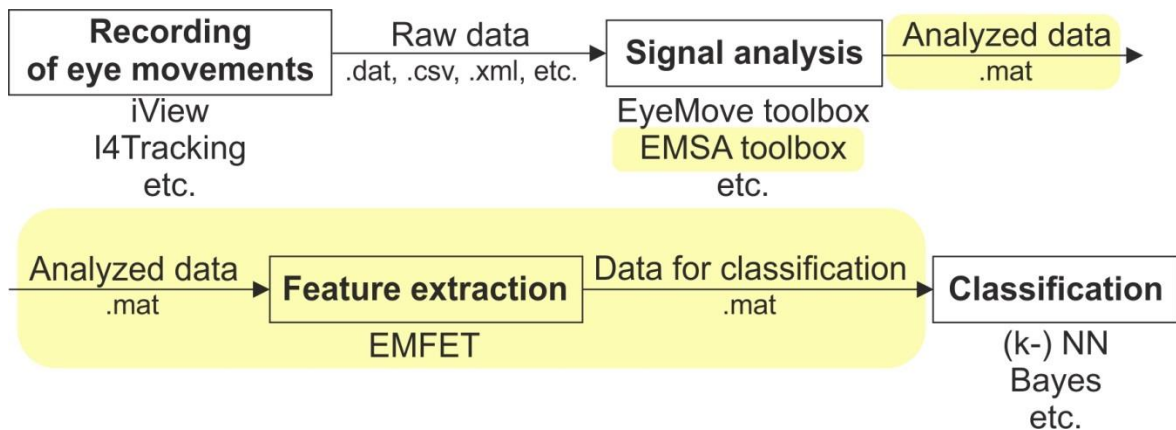


Figure 1 - A block diagram of the proposed method, the part highlighted in yellow is concerned with this bachelor thesis

## 1.1. Main goals

The main objectives of this work are:

1. Develop an application with graphic user interface for automatic mass feature extraction from results of analysis of eye movement signal. This application should be able to provide mass feature extraction for multiple analysis files, basic statistical analysis and visualization of features. For this application is important to be easily extendible in terms of additional feature extractors.
2. Design the internal structure of feature extractor and document it for facilitation of further feature extractors creation.
3. Create the mass feature extraction output files structure. These files in Matlab \*.mat file format contain the data used for classification.
4. Validate the application on provided analysis files. Apply the feature extraction from multiple input analysis files.
5. Contribute in development of Eye Movement Signal Analysis (EMSA) data structure. This data structure is developed concurrently with the application for mass feature extraction and serves as analyzed data file format.

## 1.2. Motivation

Modern technologies of videooculography allow recording the movement of human eye with the precision needed for detail examination of these movements. Several studies point the possibility to diagnose serious human conditions from the analysis of eye movement. Imagine the option of diagnose dyslexia among pre-school children even before they start to read, diagnose schizophrenia faster than symptoms develop etc. The early treatment or

different approach to this people can mean alleviation of disease symptoms and better quality of life. This possibility can be discovered in studies of eye movements.

Applications of statistical pattern recognition methods are still not common. Classic statistic methods are mostly used (e.g. hypotheses testing). Statistical pattern recognition methods can bring us different approach to the eye movements studies. To be able to use these methods, we need a tool for automatic mass feature extraction from eye movement signal analysis.

However, there is no application that can be used for feature extraction. Main objective of this thesis is to create an user application that will support mass feature extraction. As environment for this application was chosen Matlab, developed by MathWorks. It is fourth-generation programming language widely used on Department of Cybernetics, Czech Technical University. The important property of this application is its modulation character, which means that is possible to add new feature extractors and be able to process any data files (if the feature extractors are made to process this data type).

### 1.3. Recording of eye movements

The technology most widely used in the current designs of eye trackers is videooculography. It is video-based eye tracking technology using a camera with high sampling frequency. The camera focuses on one or both eyes and records their movement. The eye tracking system subsequently processes the images into raw data. This data consist of raw pupil position in coordinates (pixels or angle), pupil size and measurement details (subject details, stimuli details, etc.).

Prior to this work and recent state, there was a SMI (<http://www.smivision.com/>) infra-red videooculographical device iView 3.0 used to track eye movements of 76 children at the Department of Neurology, 2<sup>nd</sup> Medical Faculty, Charles University, Czech Republic back in year 2003.

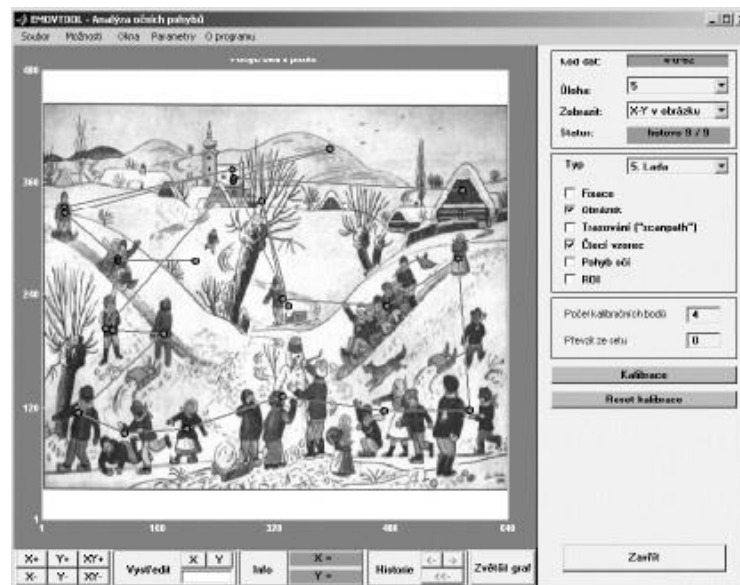
Current measurements are executed by using I4Tracking, tracking system provided by Medicton Group s.r.o. (<http://www.medicton.com/>). The measuring device consists of high speed camera attached on glasses (Picture 1), LCD screen and computer, which displays stimuli and process the data.



Picture 1 - I4Tracking measuring device

## 1.4. Signal analysis

Data measured by iView system were originally analyzed with Eyemove Toolbox (Picture 2) developed by Ing. Jakub Snopek in his diploma thesis (2003). The new data cannot be analyzed by Eyemove Toolbox and redesign of this toolbox for the new data was declined for various reasons. The source code of this toolbox was used in creating the Eye Movement Signal Analysis toolbox (EMSA toolbox) for Matlab. EMSA toolbox support data input from iView system and also I4Tracking system. Support of an additional eye tracking system is also possible.



Picture 2 - EyeMove Toolbox

## 1. INTRODUCTION

Task of the signal analysis toolbox is to process the raw data from eye tracker. This data contain a lot of artifacts, the most significant ones are those made by the eye blink. The artifacts like this needs to be found and excluded from following analysis. After the artifacts removal it is possible to analyze the parameters of eye movement, this is primary the detection of fixations and saccades, also other eye movement components if the VOG system is fast enough to record this components.

The output from signal analysis is Matlab file \*.mat consisting of raw measured data and analyzed signal data. For each measured subject is exactly one data file.

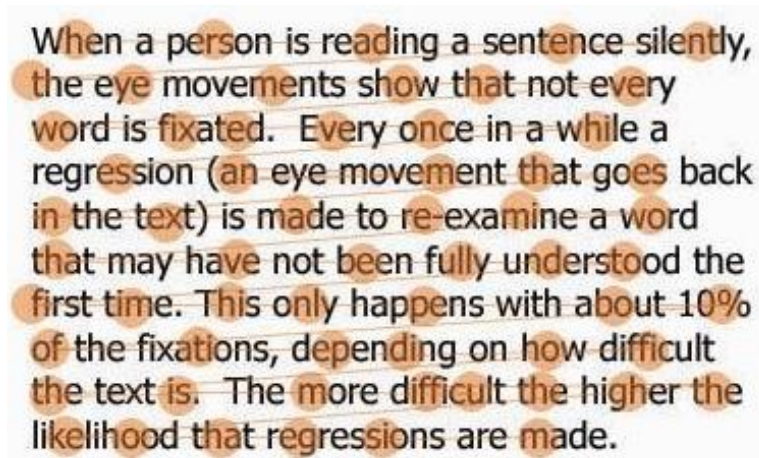
### 1.4.1. Classification of eye movements

Human eye movements consist of several types of movements. The most common types and their typical values are in Table 1. Recognition of these types is fundamental for feature extraction.

Type	Duration [ms]	Amplitude	Velocity
<b>Saccade</b>	30 - 80	4 - 20°	30 - 500°/s
<b>Fixation</b>	200 - 300	-	-
<b>Glissade</b>	10 - 40	0.5 - 2°	20 - 140°/s
<b>Smooth pursuit</b>	-	-	10 - 30°/s
<b>Microsaccade</b>	10 - 30	10 - 40'	15 - 50°/s
<b>Tremor</b>	-	< 1'	< 20'/s
<b>Drift</b>	200 - 1000	1 - 60'	6 - 25'/s

Table 1 - This table shows different types of eye movements, their length in milliseconds, amplitude and velocity (adapted from Holmqvist, 2011).

Saccades are quick, simultaneous movements of both eyes in the same direction from one fixation to another. They are the fastest movement the body can produce. The fixation happens when the eye remains still over a period of time (for example when eye stops on a word when reading - Picture 3). Glissade is a post-saccadic movement of the eye, when the eye „wobbles“ a little before coming to a stop. Smooth pursuit is when the eye follows a moving object. It is driven by different part of brain than saccades. Moving stimuli is required for smooth pursuit. Microsaccades are very short movements of the eye that are trying to bring the eye back to the center of fixation (for example after drift). Tremor is a small movement of frequency around 90Hz, whose exact role is unclear (it can be imprecise muscle control). Drifts are slow movements taking eye away from the center of fixation.



Picture 3 - Saccades and fixations during reading task (picture adapted from Rayner, 2007)

## 1.5. Feature extraction

The feature extraction is a way to statistically evaluate the recorded data and compare the results between the tested subjects. The methods can be divided into four main groups: position based features, numerosity based features, latency and distance based features and frequency based features. The result of feature extraction can be numerical value, string, vector of values or strings. For the mass feature extraction are the results stored in matrixes for simplification of further classification or data mining.

### 1.5.1. Position based features

The most used feature extractors are based on position of eye movement events. The position duration measures and input/output directions are significant. Some of the position based feature extractions are prior to numerosity based feature extractions, for example determining the regressions (quick backward saccades).

The mean durations of eye movement events are mostly used in all researches. The mean fixation duration is considered as an indication of visual-information-processing time. This duration can refer to depth of understanding the text.

### 1.5.2. Numerosity based features

To count the eye movement events is one of the methods to quantify them. It can be expressed in absolute numbers (e.g. how many times saccades occurred during certain task), in proportion to the total number of events or as rate over time. There is often little point to calculate the total number of saccades or any other event in the whole recorded

data. Limiting the counting process on specified tasks is necessary. As exception is for example the blink rate.

In reading tasks is frequent counting the number of fixations per line or word. Number of fixations in the area of interest is called fixation density (J.M. Henderson, 1999). According to review in Jacob and Karn (2003) number of fixations is one of the most used metrics in usability research. In study of Rubino and Minden (1973) authors found that children with learning disabilities made significantly more fixations, but there were no significant differences in fixation's duration. Also the number of regressions during reading is one of the most major factors in diagnose of genetic and developmental disorders (Gilbert, 1953; Pavlidis, 1985; Biscaldi, 1998).

With still images stimuli the number of saccades should be equal to the number of fixations. For stimuli that elicit smooth pursuit we can count the saccadic rate, it would be a measure of prevalence of catch-up saccades. Studies reviewed by O'Driscoll and Callahan (2008) tend to show that participants with schizophrenia have the saccadic rate much higher than control group subjects. Number of undesirable fixations during smooth pursuit is also a common feature.

### **1.5.3. Latency and distance based features**

Latency is a measure of time delay between events, often can be used as reaction time based features. Most of latency based features are used in dynamic tasks when there are new objects flashing on the screen and the time to saccadic start is measured. Also eye-voice latency is measured or pupil dilation latency after an event that start the dilation (e.g. bright light).

Distance based features are not so much often. The eye-mouse distance is used in some tracking systems measuring the coordination between hand and eye. If tracking system support tracking of left and right eye simultaneously, the distance between points of gaze of each eye can be measured.

### **1.5.4. Frequency based features**

Study of the power spectral density of eye movement signal (Schmeisser, 2001) indicates that frequency based features has significant meaning during reading tasks. Eye movements have two components – horizontal and vertical. Frequency analysis can be used to find some significant properties of both signals (Macaš, 2005).



## 1.6. Classification

Classification is based on the output of feature extraction. There are many types of statistical classification methods. In many cases there are two stages of classification: the learning stage and the classification stage. In the learning stage are training sets of data presented to classifier, based on this data classifier can decide during the classification stage to which set of categories a new observation belongs. Generally, the more of the training data, the more accurate is the result of classification.

Classification of eye movements usually deals with subgroups of people with some disorders (dyslexia, schizophrenia, etc.) and healthy test group. As classifiers are mostly used k-Nearest Neighbor, Bayes classifier or neural networks.

## Chapter 2

# Implementation

In this chapter is described the eye movement signal input data, their conversion by Eye Movement Signal Analysis (EMSA) toolbox for Matlab into EMSA files and structures of EMSA and feature extractors. For mass feature extraction was created the application Eye Movements Feature Extraction Tool (EMFET), which can be used for mass feature extraction of any Matlab structure file with adapted feature extractors.

### 2.1. Signal data

Analyzed signal data by EMSA toolbox were provided in \*.mat files readable in Matlab. These data contains raw measured data, calibration configuration, subject information, processed signal data etc. Because understanding of EMSA structure is required for feature extractor creation and no one yet described it, description of EMSA structure is necessary.

#### 2.1.1. EMSA structure

The EMSA files are Matlab variables of type STRUCT. This variable type is similar to Java Object, but has different syntax. I personally contributed on the development of EMSA structure. Figure 2a and Figure 2b shows the complete structure of EMSA file as is to date 2013/05/21. Further development of EMSA structure is possible and there can be major structure changes. For the latest form of EMSA structure please refer to development coordinator Ing. Martin Macaš.

## 2. IMPLEMENTATION

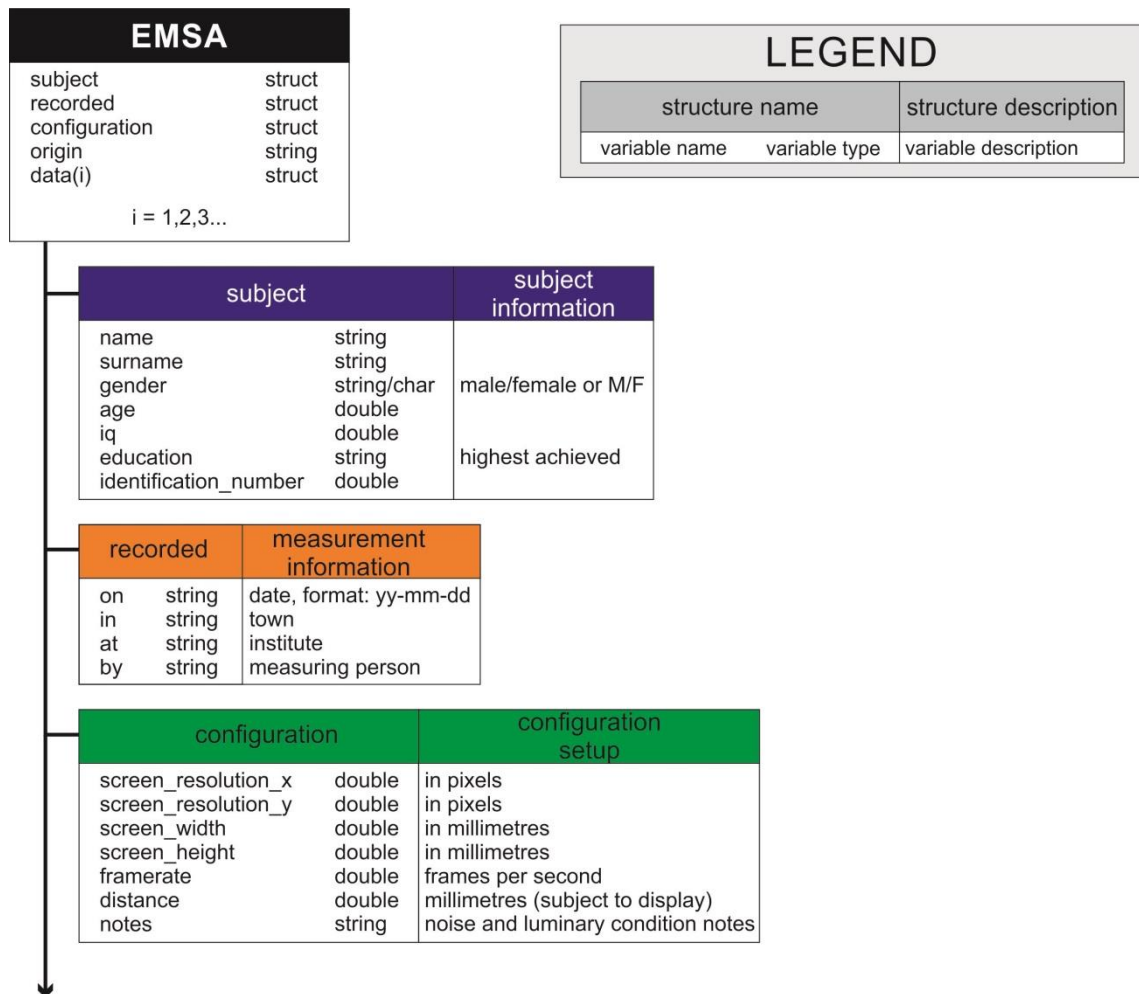


Figure 2a - EMSA structure

## 2. IMPLEMENTATION

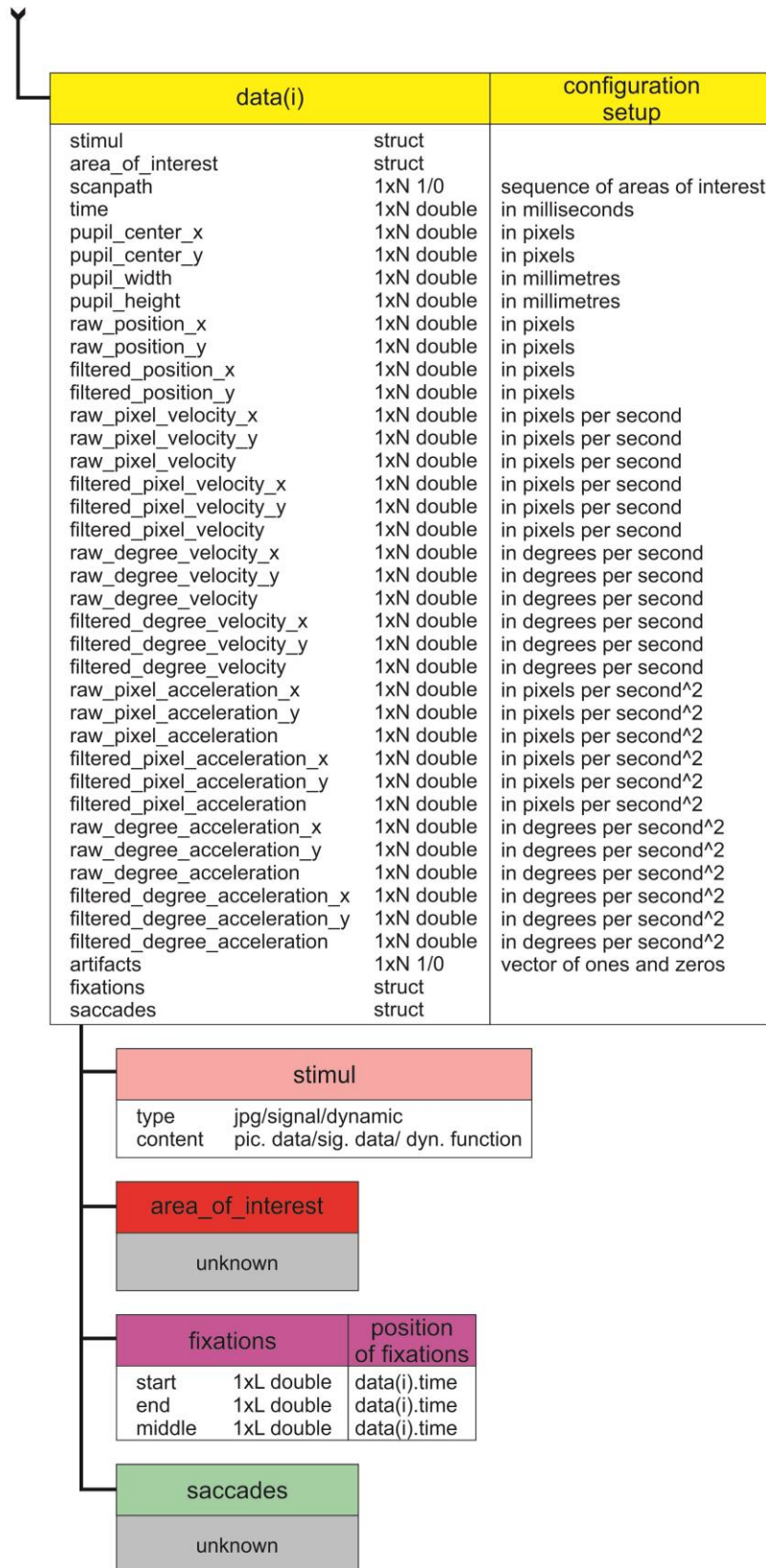


Figure 2b - EMSA structure

## 2.2. Feature extractors

In fact the feature extractor is Matlab function M-File that has specific structure. The structure is well described in the file `extractor_sample.m` stored on attached CD. This function can have multiple numeric input arguments used for internal calculations. These values can specify offset, sampling frequency, accuracy etc. It is important to abide the internal structure of feature extractor for the smooth running of feature extraction.

It is highly recommended to provide the calculation part of feature extractor with try-catch blocks. In case of some calculation failure or missing input data in try block, handle the error in catch block and provide the right output or NaN.

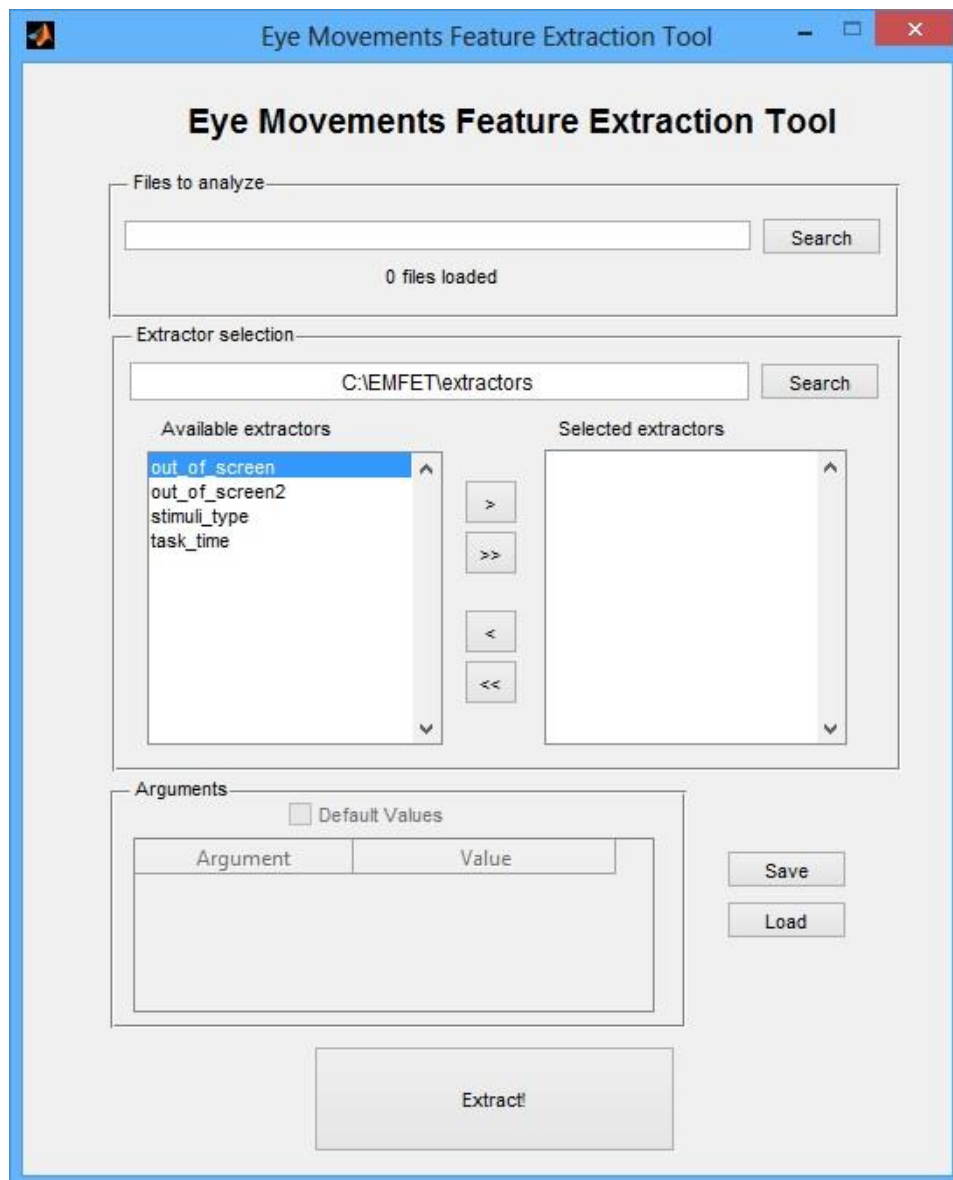
The feature extractor always counts with only one EMSA file (or any other Matlab data structure) and it is always the first input argument. If the feature extractor has special input arguments, they must be specified in arguments cell array. The special input arguments must have numeric value (use 1 or 0 for Boolean values).

Output of the feature extractor is 1xN cell array. This cell array can contain numbers, strings, or other Matlab structures. Count N must be always same for one feature extractor, the same feature extractor cannot give for one data file 1x3 cell array and for another data file 1x5 cell array.

## 2.3. Graphical User Interface

Because feature extractor has as input only one analysis file, the Eye Movements Feature Extraction Tool (EMFET) was made as tool for mass extraction. This application (Picture 4) was made as Matlab Graphical User Interface (Matlab GUI) and needs Matlab 2009 or newer in order to run properly.

EMFET supports as input data any \*.mat Matlab structure, but the feature extractors must be adapted on this structure.



Picture 4 - Application window

EMFET support multiple analysis data files extraction by multiple feature extractors. The application window is separated into 3 panels: in the first panel (Files to analyze) are analysis files selected, in the second panel (Extractor selection) is specified the folder with feature extractors and then selected which feature extractors will be used for feature extraction, the third panel (Arguments) is there to set the individual arguments for each feature extractor. The selected feature extractors and their arguments settings can be saved or load.

## 2. *IMPLEMENTATION*

---

Everything about the feature extraction and EMFET application is described in Application Manual, which is in Appendix.

### **2.3.1. Output**

The application output is 3 cell arrays: Result, Extractors and Data. Result is MxN cell array (M is total number of files to analyze, N is sum of selected feature extractors outputs) containing the feature extraction results. Extractors is 1xN cell array containing the feature extractors names, input arguments (if feature extractor has any). Data is Mx1 cell array containing the names of data files.

There are 3 options how to save the computed results and use them in Matlab, for more information see Application Manual (chapter Saving the result), which is in Appendix.



## Chapter 3

### Validation

To test the application it was necessary to create feature extractors. Because the EMSA structure is still in development stage and the methods for filtering the signal and feature detection are not available at this time, the extractors for feature extraction had to be made for the raw data from the tracking system. These feature extractors can be used for checking the accuracy of the measurements and calibration of the camera, also can be used in data mining.

The records of eye movement signals measured by I4Tracking system were used for feature extraction in this work. They were partially processed by EMSA toolbox and saved as Matlab \*.mat files. Each file corresponds to one measured subject, whose eye movements have been recorded and analyzed. All these fifteen subjects were without eye disorders and data serves as test data for EMSA toolbox and feature extraction creation.

#### 3.1. Feature extractors description

##### 3.1.1. Out of screen proportion (out\_of\_screen, out\_of\_screen2)

The recorded data (raw\_position\_x, raw\_position\_y) are taken from the tracking system. How accurate the data are depends on the conditions of measurement and tracking system abilities. Mostly the head of the subject is not firmly attached (to be not disturbed and feel more naturally). When the subject moves the head and there is no compensation from the side of the tracking system, then the data move off the screen into negative values

### 3. VALIDATION

or values higher than screen resolution. The perfect example is on Figure 3, the tracking system in this case I4Tracking does not compensate the movement of the head and part of data is out of screen.

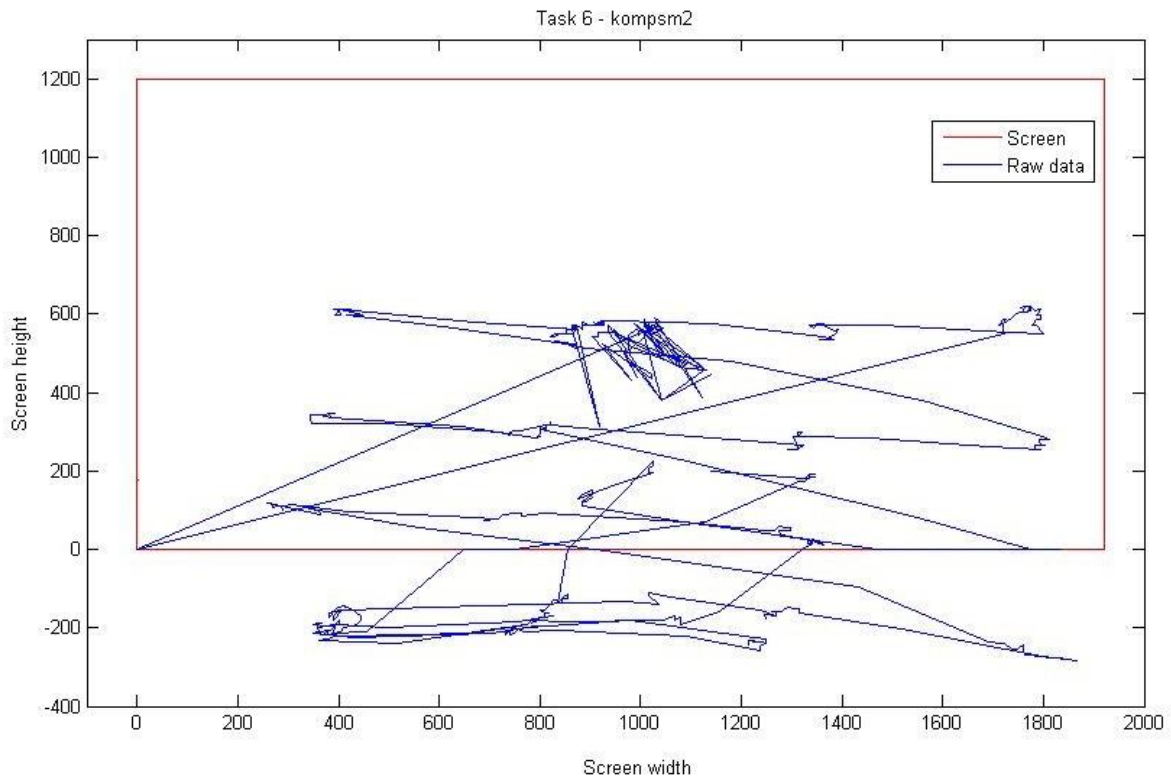


Figure 3 - 33.4% of signal data are out of screen

This can cause issues in further analysis of data or feature extraction. Also the stimuli picture when given under the data as background does not correspond with the data, which can be very confusing.

If the tracking system automatically compensate the head movement, or the compensation is done by EMSA toolbox analysis, there can still be some out of screen data. For example when the subject just look outside the screen (e.g. on measurement supervisor or some distraction).

The feature extractors `out_of_screen` and `out_of_screen2` gives the percentage of data outside of screen in range 0-100%. If there is no compensation of the head movement, the higher this number is the more movement of head this subject made. But if there is the compensation of the head movement, the higher this number is the more distracted was the subject during the measurement and this measurement can be considered as faulty.

### 3. VALIDATION

The out\_of\_screen feature extractor considers all tasks in file. Instead out\_of\_screen2 consider only one task in file, the task number is given as argument and is 1 in default.

#### 3.1.2. Stimuli type (stimuli\_type)

In comparison of feature extraction from two and more data files is often essential that the extracted features are from the same stimuli. This feature extractor lists the stimuli type of task number given as argument (default 1).

#### 3.1.3. Task time (task\_time)

In most cases the stimuli have not specified projection duration and the subjects tell the supervisor when they are done with the task. The time they needed for task is feature, that can help in diagnose and assessment if the measurement is successful. The example how task time can differ by subjects is plotted in Figure 5. All these subjects had to make the same task called dots2 (Figure 4). The stimulus for task dots2: *Look through the dots one by one and knock by hand when you will be over.* The results are in milliseconds.

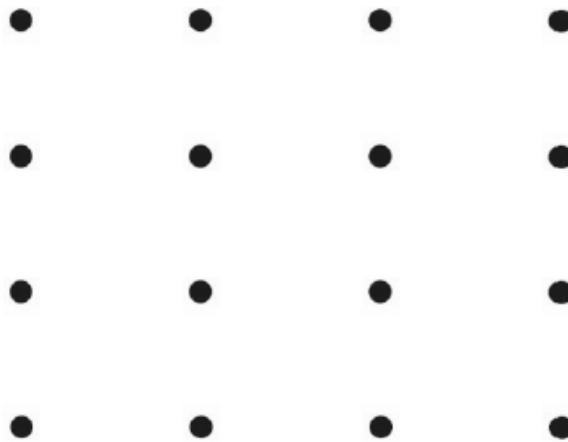


Figure 4 - dots2

## 3. VALIDATION

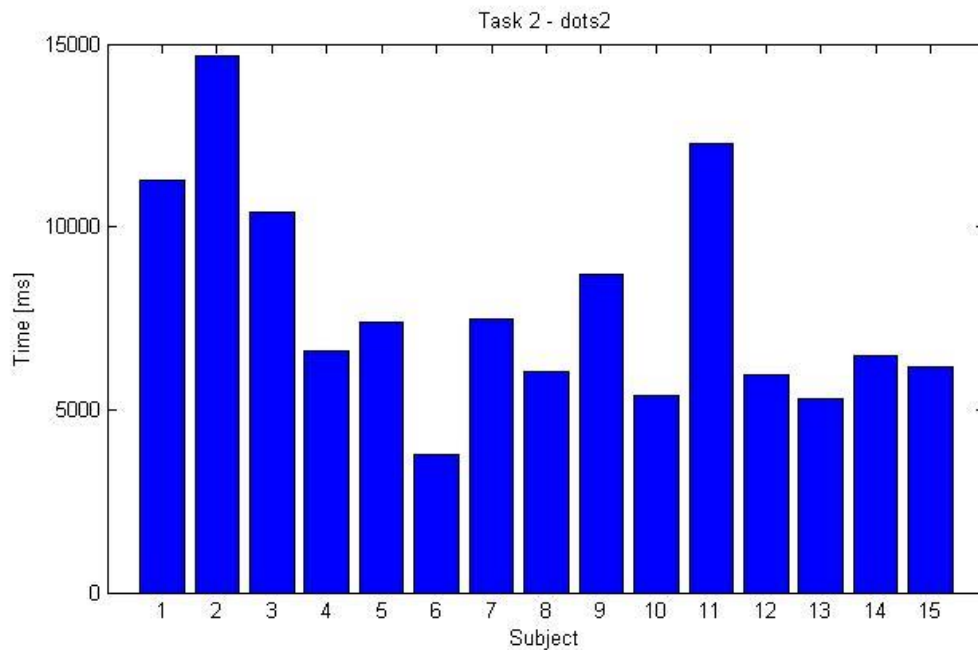


Figure 5 - Task time of different subjects

## 3.2. Mass feature extraction

In this section is the mass feature extraction tested. For the capability of mass feature extraction the application Eye Movements Feature Extraction Tool (EMFET) was created. As the input were used the 15 data files in EMSA structure measured by I4Tracking system. The output of the test feature extraction is shown below in Tables 2-4.

Table 2 saved as variable *extractors* is where the feature extractors names and arguments are stored. The structure of the cell content is: 'feature extractor name'\_'argument values'/'index'. Feature extractor name is always used, argument values are used only if feature extractor support variable arguments, index is used only if one feature extractor returns more than one result (in that case each result is in one cell).

out_of_screen	out_of_screen2_6	stimuli_type_6	task_time_6	task_time_2
---------------	------------------	----------------	-------------	-------------

Table 2 - extractors

### 3. VALIDATION

result20121213_152458.mat
result20121213_153917.mat
result20121213_154811.mat
result20121213_155653.mat
result20121213_161159.mat
result20121213_161924.mat
result20121213_163340.mat
result20121213_164358.mat
result20121213_165218.mat
result20121213_170206.mat
result20121213_171534.mat
result20121213_173252.mat
result20121213_174750.mat
result20121213_175426.mat
result20121213_175844.mat

Table 3 - file\_names

0.19	0	Kompsm2	21316	11266
7.03	33.37	Kompsm2	16289	14663
0.01	0	Kompsm2	9735	10416
1.48	0	Kompsm2	8661	6606
0.20	0	Kompsm2	8661	7387
0.02	0	Kompsm2	5820	3791
0	0	Kompsm2	10837	7487
0	0	Kompsm2	9178	6038
3.71	15.95	Kompsm2	9179	8703
0	0	Kompsm2	7429	5390
0	0	Kompsm2	8203	12281
0	0	Kompsm2	6641	5955
0	0	Kompsm2	6936	5289
0.02	0	Kompsm2	6385	6472
0.70	0	Kompsm2	6654	6171

Table 4 - results

Table 3 saved as variable *file\_names* is where the input data filenames are stored. Table 4 is the *results* table, into this table is stored the results of extraction. If Table 2 has N cells and Table 3 has M cells, the size of Table 4 is MxN, in other words the size of the *results* table is number of extractors' results times number of input files.

## Chapter 4

### Conclusion

Application for feature extraction from eye movement signal analysis was proposed in this work. This application manages mass feature extraction from one or more input data files in Matlab format. The format of input data is independent on Eye Movements Feature Extraction Tool (EMFET) application. Output files from this application are in Matlab file format, that can be used for classification, diagnose or in data mining. User manual for this application is attached in Appendix.

The important part of this thesis is design of the feature extractor structure. Feature extractors are Matlab single-function M-Files automatically recognized by EMFET application with designated structure (structure is described in Chapter 2.2. and in file extractor\_sample.mat attached on CD). They have one mandatory argument, which is the single input analyzed data. From this is evident the 1:N relation between the input data structure and the feature extractor. The feature extractors can have extra variable arguments that are set before each feature extraction in the graphical user interface. EMFET is capable of saving and loading the settings of selected feature extractors and their variable arguments.

For testing purposes were created 4 feature extractors (they are described in Chapter 3.1.) for EMSA files. The complete structure of EMSA file is described in Chapter 2.1.1. for further feature extractor creation. As input data served 15 measurements of healthy subjects measured via I4Tracking system. They were partially processed by Eye Movement Signal Analysis toolbox. EMFET application were used for the mass feature extraction from these measurements, thereby was verified the functionality in practice.

### 4.1. Future work

There is a lot of space for the future work. At first creating more feature extractors is necessary for a successful classification. The feature extractors based on features described in Chapter 1.5. would be appropriate to create for EMSA files.

The EMFET application can be redesigned as server-client application or web application. This would cause end of problems with Matlab licensing, Matlab would be only on server machine with single Matlab license. All the clients will use the numerous possibilities of server machine.

Improving the EMSA toolbox internal methods (this is related to EMSA structure) is also possible. The current status is not sufficient for studying of eye movements and there need to be a lot of improvements.

Figure 1 shows the system of eye movement data acquisition and their processing, the blocks are separated as well as the software for analysis of eye movement. In one day this could be all connected in one big application, which will do everything.



The background of the entire page is a solid blue color. At the top, there are several vertical columns of white binary code (0s and 1s) that appear to be falling or scrolling down, similar to the 'Matrix' effect. In the lower half of the image, there is a close-up, high-contrast photograph of a person's eyes, looking slightly upwards and to the right. The eyes are dark and have long, dark eyelashes. The overall aesthetic is technological and futuristic.

# Eye Movements Feature Extraction Tool EMFET

---

Version 1.0

**SOFTWARE MANUAL**

# Table of Contents

**INTRODUCTION..... 24**

**STARTING THE APPLICATION..... 24**

Set Matlab search path ..... 24

Start the application ..... 25

**EXTRACTION CONFIGURATION..... 26**

Select the extraction data..... 26

Load and select feature extractors..... 26

Set parameters of feature extractors ..... 27

Save/Load the configuration ..... 27

**EXTRACTION ..... 27**

Saving the result ..... 28

## Introduction

This application the Eye Movements Feature Extraction Tool (EMFET) was created to simplify the extraction of eye movements symptoms.

To run this application you need Matlab version 2009 or newer.

## Starting the application

### Set Matlab search path

There are two ways to add the application folder to a Matlab search path, use one of these:

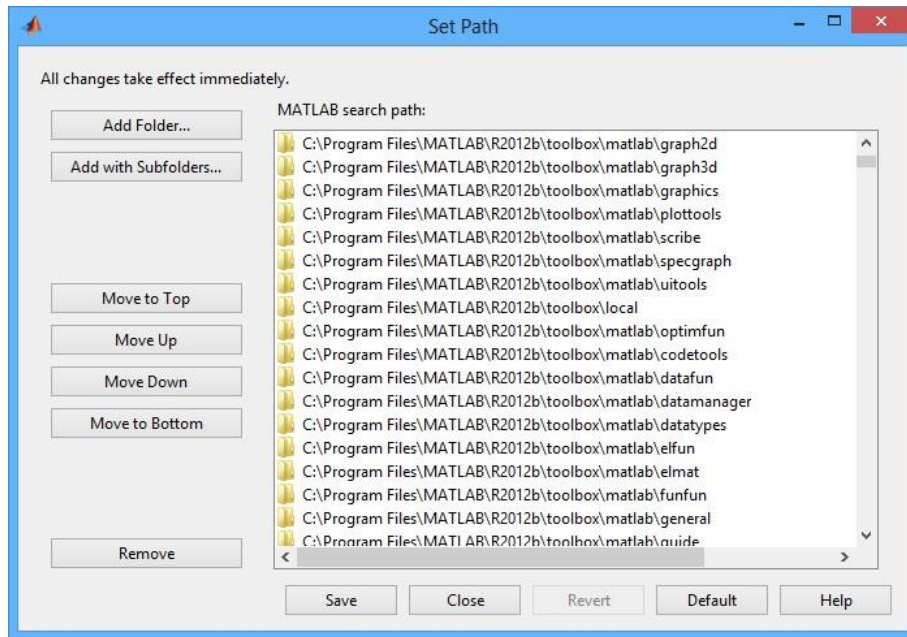
1. Set your current Matlab folder to folder with EMFET. To check your current Matlab folder type into Matlab command window:

```
>> pwd  
  
ans =  
  
C:\EMFET
```

2. Add EMFET folder to Matlab search path. You can do it temporarily by typing into Matlab command window:

```
>> path (path,'*folder*')  
  
Note: change *folder* to EMFET folder, for example  
C:/EMFET
```

or permanently using the Matlab Set Path dialog box (use the Add Folder button) as you see on [Picture 1](#).



Picture 1 - Matlab Set Path dialog box

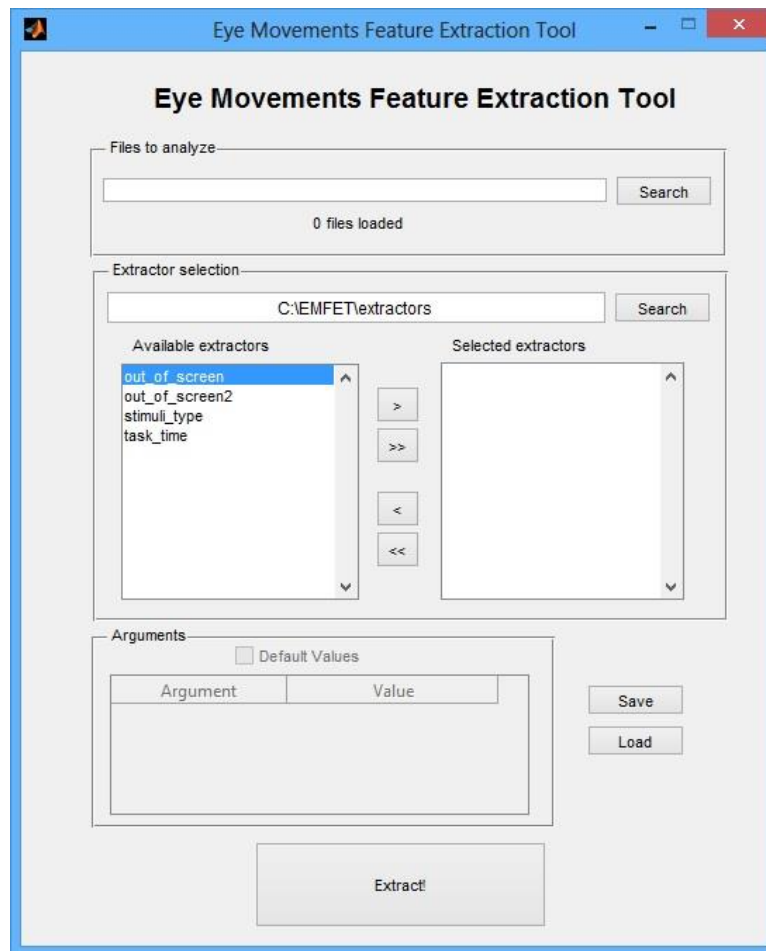
## Start the application

Start the EMFET application from the Matlab command window, the syntax is:

```
>> emfet
>> result = emfet
>> [result extractors] = emfet
>> [result extractors data] = emfet
```

`[result extractors data] = emfet` saves the last computed results of application into Matlab workspace when you quit application. There are several other ways how to import the result into Matlab workspace, see chapter [Saving the result](#).

The application will start in new window as you see on [Picture 2](#).



Picture 2 - Application window

## Extraction configuration

### Select the extraction data

Press the Search button in panel Files to analyze and locate EMSA files (or any other Matlab structure \*.mat files) in your computer. You can choose multiple files in one folder (use the mouse selection or CTRL+click to select more than one file).

### Load and select feature extractors

The default folder for feature extractors is extractors subfolder in EMFET folder. If you want to use other feature extractors then copy the feature extractor M-files to this folder and restart the application, or use the Search button in panel Extractor selection and locate the folder with your feature extractors M-files in your computer. Feature extractors then appear in Available extractors' listbox. Files in the chosen folder that are not extractors are displayed on Matlab Command Window.

Using the arrow buttons select feature extractors you want to use on selected data files. The double arrow button adds all feature extractors from one listbox to another. If you want to

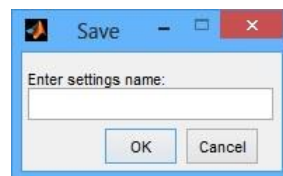
use one extractor twice (or more), for example with another parameter setting, simply add more same extractors in the Selected extractors listbox.

### Set parameters of feature extractors

If the feature extractor supports variable parameters values, you can set them in panel Arguments. In the listbox Selected extractors highlight the feature extractor you want to modify, uncheck Default Values checkbox and modify the parameters. This can be done for all selected feature extractors.

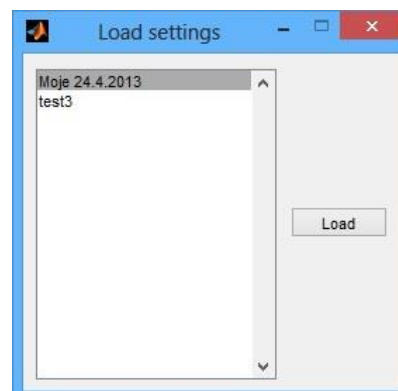
### Save/Load the configuration

Application allows saving the selection of extractors and their parameter settings. Once you have selected the feature extractors and modified the parameters press the Save button, the Save dialog appears ([Picture 3](#)), enter the settings name (for example “My settings 18.4.2013”) and confirm the save with OK button. The saves will remain even if you close the program and Matlab.



Picture 3 - Save dialog

Before loading the configuration settings make sure that you have feature extractors you want to load in Available extractors listbox (if not, see [Load and select feature](#) extractors). Press the Load button, the Load dialog appears ([Picture 4](#)), choose the one you want to load and press Load button.



Picture 4 - Load dialog

## Extraction

Once you have selected the files to analyze and extractors you want to use press the Extract button. Matlab automatically calculate the result cell arrays and save dialog will appear.

This is one of three options how to save computed results. If you cancel this save dialog, you can still save the result using the other two methods.

### **Saving the result**

As mentioned before there are 3 options how to save computed results and use them in Matlab:

1. Save variables using save dialog, that appears with [Extraction](#). Variables are stored in \*.mat file, which can be easily load into Matlab workspace, see Matlab load function.
2. Start the application using syntax mentioned in chapter [Start the application](#). Then when you quit the application, Matlab will store the last computed results into variables you defined.
3. After every extraction there is automatically saved the results into file last\_result.mat in the folder with EMFET.



# Bibliography

- Biscaldi, M.; Gezeck, S.; Stuhr, V. (1998). Poor saccadic control correlates with dyslexia. *Neuropsychologia*, 36 (11), 1189-1202.
- Gilbert, C.L. (1953). Functional motor efficiency of the eyes and its relation to reading. *University of California Publications in Education*, (11), 159–231.
- Henderson, J.M.; Hollingworth, A. (1999). The role of fixation position in detecting scene changes across saccades. *Psychological Science*, 10, 438-443.
- Holmqvist, K.; Nyström, M.; Anderson, R.; Dewhurst, R.; Jarodzka, H.; van de Weijer, J. (2011). *Eye tracking, A comprehensive Guide to Methods and Measures*.
- Jacob, R.J.K.; Karn, K.S. (2003). Commentary on Section 4 – Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises. *The Mind's Eye, Cognitive and Applied Aspects of Eye Movement Research*, 573-605.
- Macaš, M. (2005). *Dyslexia Detection Using Artificial Neural Networks*. Diploma thesis.
- O'Driscoll, G.A.; Callahan, B.L. (2003). Smooth pursuit in schizophrenia: A meta-analytic review of research since 1993. *Brain and Cognition*, 68 (3), 359–370.
- Pavlidis, G.Th. (1985). Eye movements in dyslexia - their diagnostic significance. *Journal of Learning Disabilities*, 18(1), 42–49.
- Rayner, K.; Castelano, M. (2007). Eye movements, *Scholarpedia*, 2(10), 3649.
- Rubino, A.C.; Minden, A.H. (1973). An analysis of eye movements in children with reading disability. *Cortex*, (9), 217–220.
- Schmeisser, E.T.; McDonoug, J.M.; Bond, M.; Hislop, P.D.; Epstein, A.D. (2001). Fractal analysis of eye movements during reading. *Optometry and Vision Science*, 78(11).805–814.
- Snopek, J. (2003). *Metody analýzy záznamu očních pohybů při čtení a v sekvenčních úlohách*. Diploma thesis.